Using Abstraction in Multi-Rover Scheduling

Bradley J. Clement and Anthony C. Barrett
Artificial Intelligence Group
Jet Propulsion Laboratory
{bclement, barrett}@aig.jpl.nasa.gov

Motivation

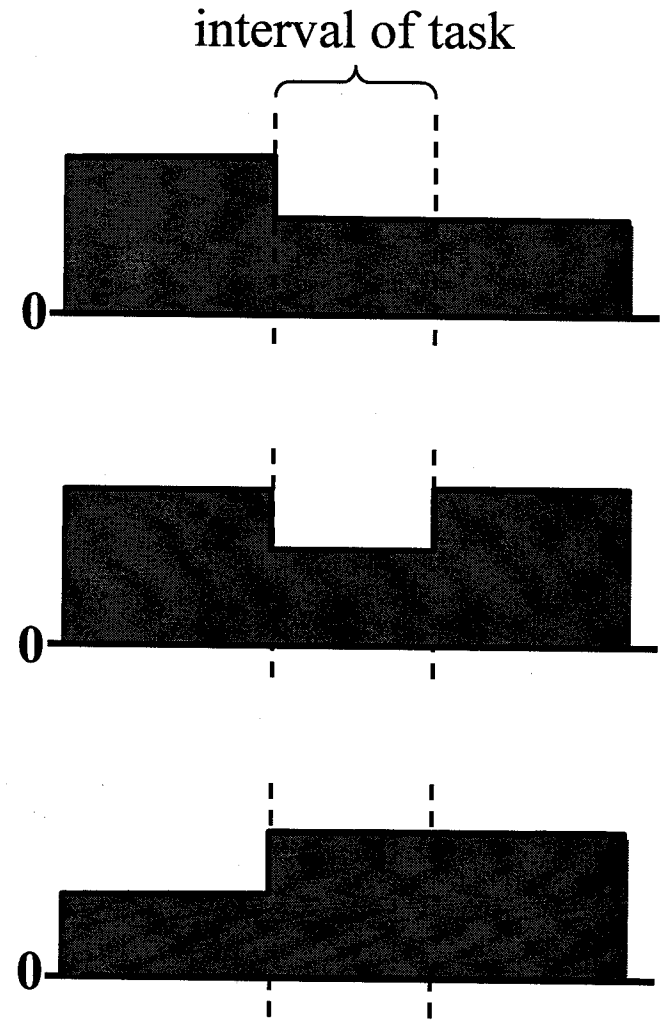
- Current trends within NASA programs point toward a need to coordinate flight projects to:
  - manage shared resources or
  - generate multiple sensor science products.
- Operations staffs must coordinate the schedules of these interacting spacecraft (or instruments).
- Reasoning about schedules at abstract levels offers performance advantages in resolving schedule coordination conflicts.
- Resolving conflicts at abstract levels preserves choices in plan refinement for flexible execution.

Contributions

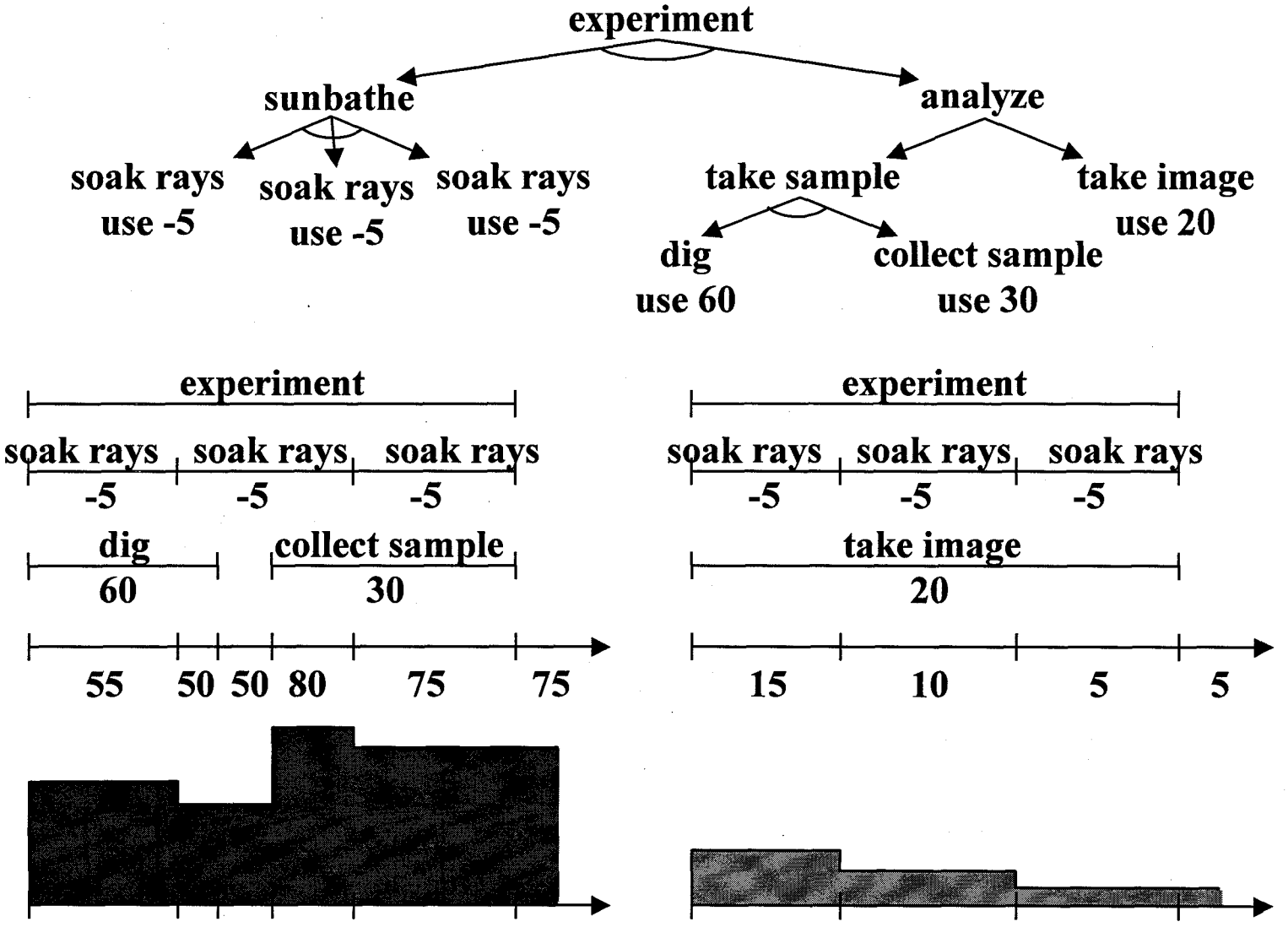
- Algorithm summarizing metric resource usage for abstract activities
- Complexity analysis showing that iterative repair scheduling operations are exponentially cheaper at higher levels of abstraction when summarizing activities results in fewer constraints and temporal constraints
- Experiments in a multi-rover domain that support the analysis
- Comparison of search techniques for directing the refinement of activities in an iterative repair planner that show how summary information can further improve performance in finding solutions

Resource Usage

- Depletable resource
  - usage carries over after end of task
  - *gas* = *gas* - 5
- Non-depletable
  - usage is only local
  - zero after end of task
  - *machines* = *machines* - 2
- Replenishing a resource
  - negative usage
  - *gas* = *gas* + 10
  - can be depletable or non-depletable



Summarizing Resource Usage

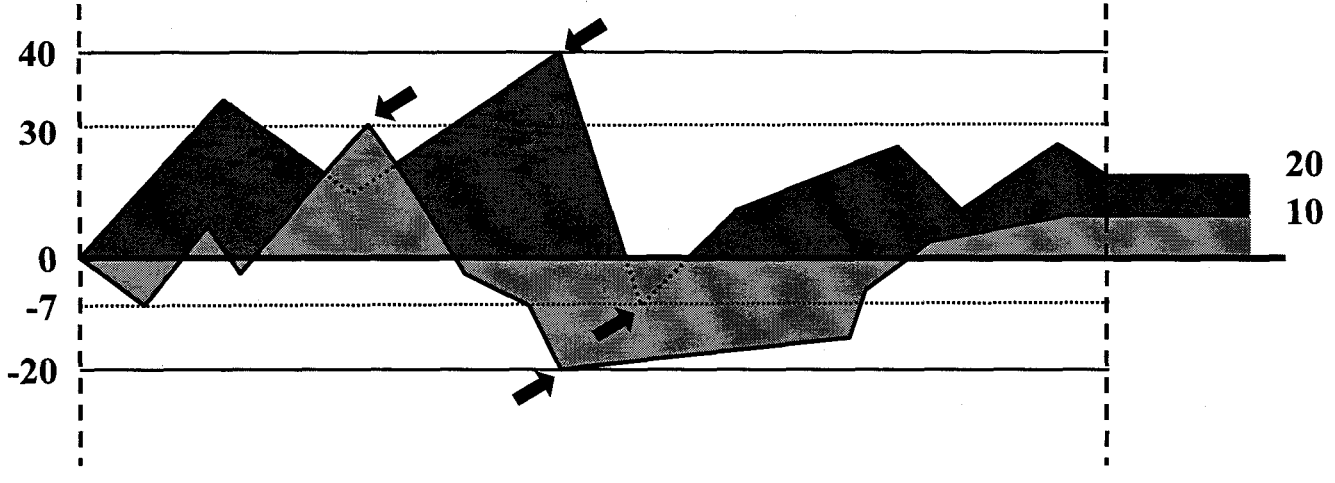


Battery energy usage for two decompositions

Summarizing Resource Usage

summarized resource usage ≡

< local_min_range, local_max_range, persist_range >



< [-7, -20],[30, 40],[10, 20] >

Captures uncertainty of decomposition choices and temporal uncertainty of partially ordered actions

Resource Summarization Algorithm

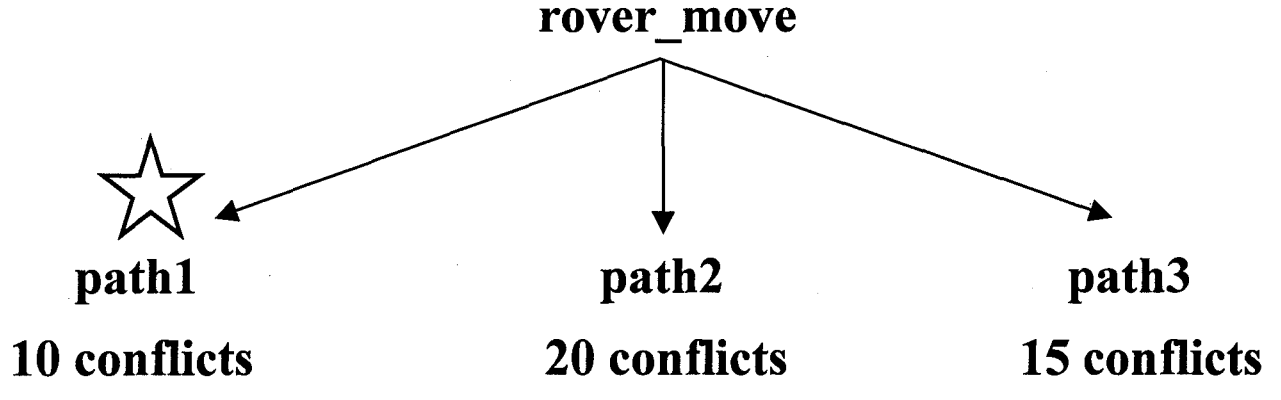
- Can be run offline for a domain model
- Run separately for each resource
- Recursive from leaves up hierarchy
- Summarizes parent from summarizations of immediate children
- Considers all legal orderings of children
- Considers all subintervals where upper and lower bounds of children's resource usage may be reached
- Exponential with number of immediate children, so summarization is really constant for one resource and $O(r)$ for r resources

Decomposition Strategies

- Expand most threats first (EMTF)
  - instead of moving activity to resolve conflict, decompose with some probability (decomposition rate)
  - expands activities involved in greater numbers of conflicts (threats)
- Level expansion
  - repair conflicts at current level of abstraction until conflicts cannot be further resolved
  - then decompose all activities to next level and begin repairing again
- Relative performance of two techniques depends decomposition rate selected for EMTF

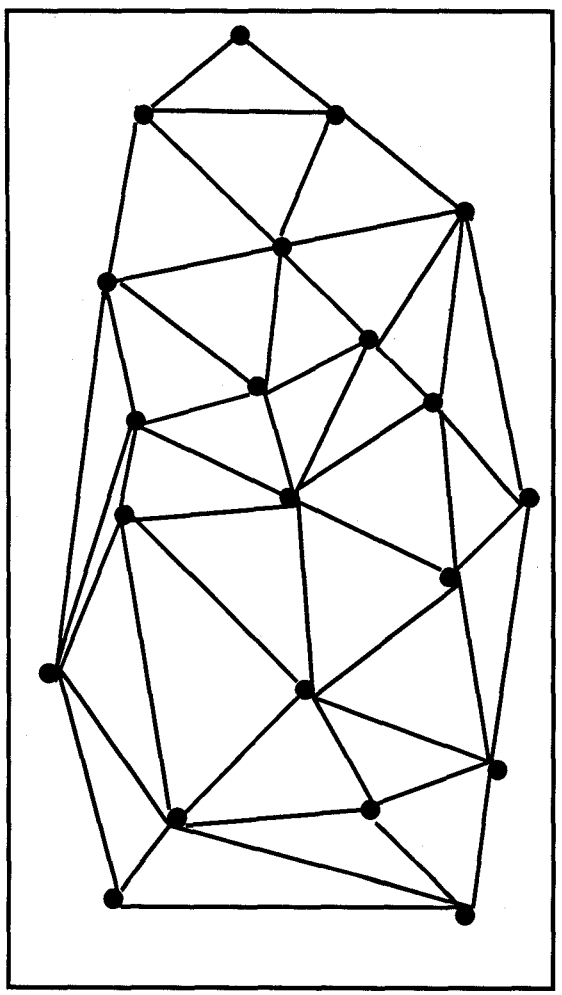
Decomposition Strategies

- FTF (fewest-threats-first) heuristic tests each decomposition choice and picks those with fewer conflicts with greater probability.



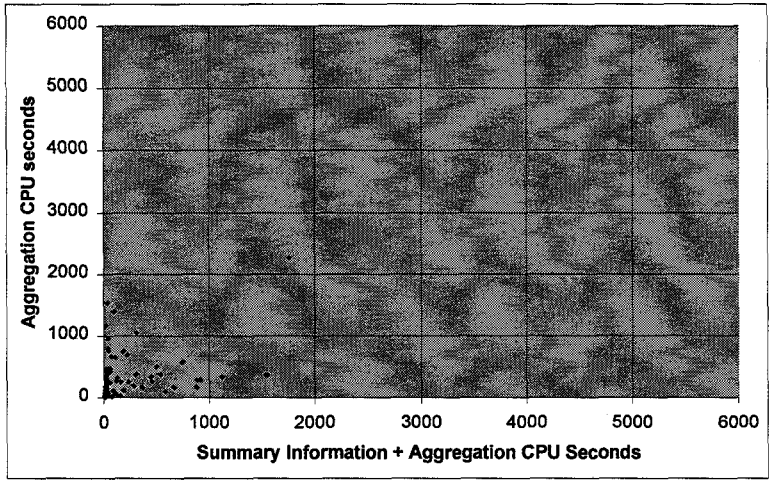
Multi-Rover Domain

- 2 to 5 rovers
- Triangulated field of 9 to 105 waypoints
- 6 to 30 science locations assigned according to a multiple travelling salesman algorithm
- Rovers' plans contain 3 shortest path choices to reach next science location
- Paths between waypoints have capacities for a certain number of rovers
- Rovers cannot be at same location at the same time
- Rovers cannot cannot cross a path in opposite directions at the same time
- Rovers communicate with the lander over a **shared** channel for telemetry--different paths require more bandwidth than others

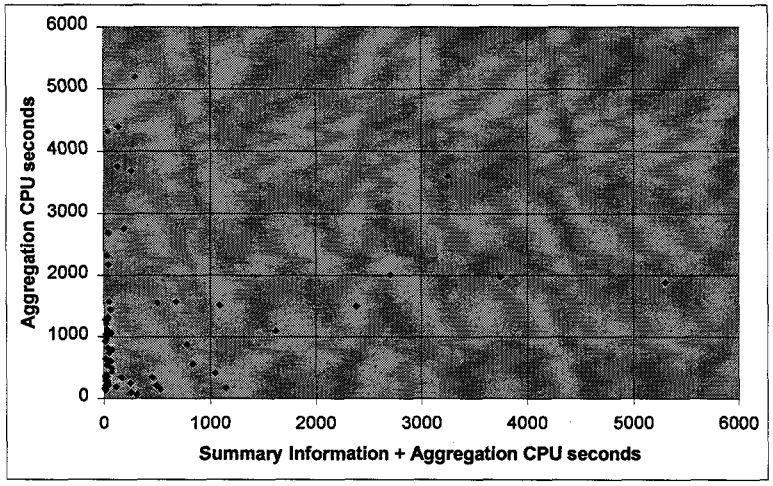


Experiments in ASPEN for a Multi-Rover Domain

- Performance improves greatly when activities share a common resource.

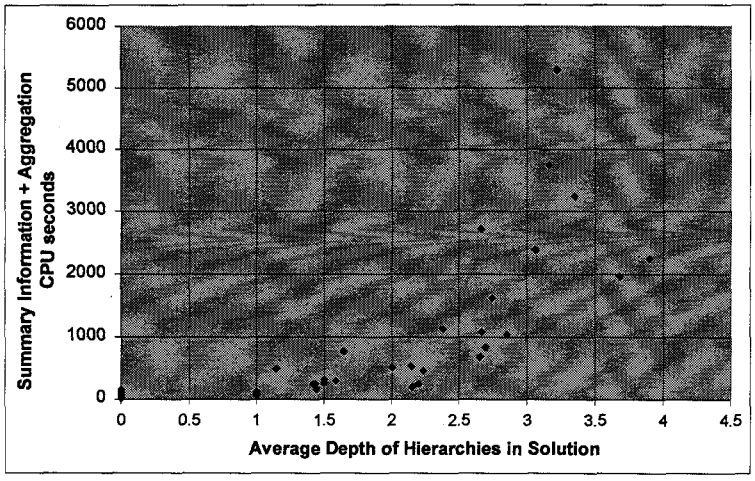


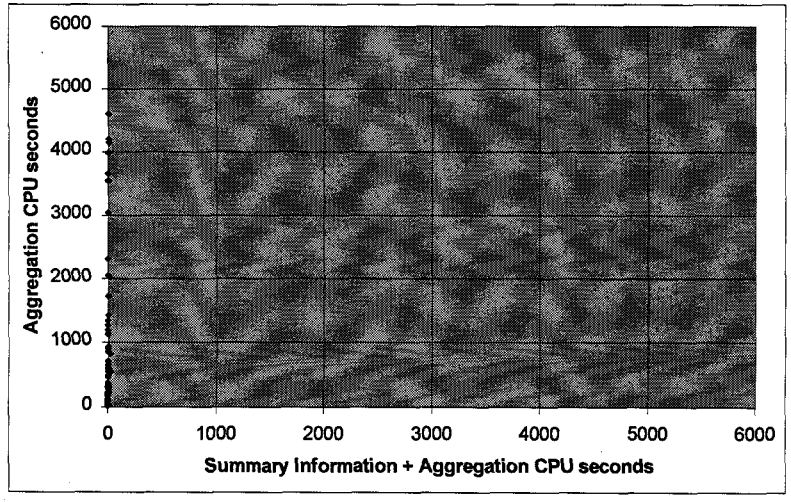
Rarely shared resources (only path variables)



Mix of rarely shared (paths) and often shared (channel) resources

- CPU time required increases dramatically for solutions found at increasing depth levels.

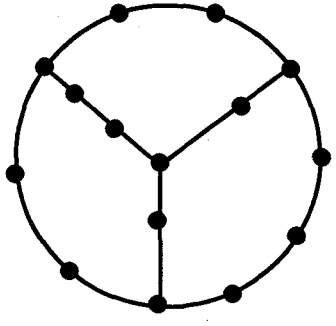




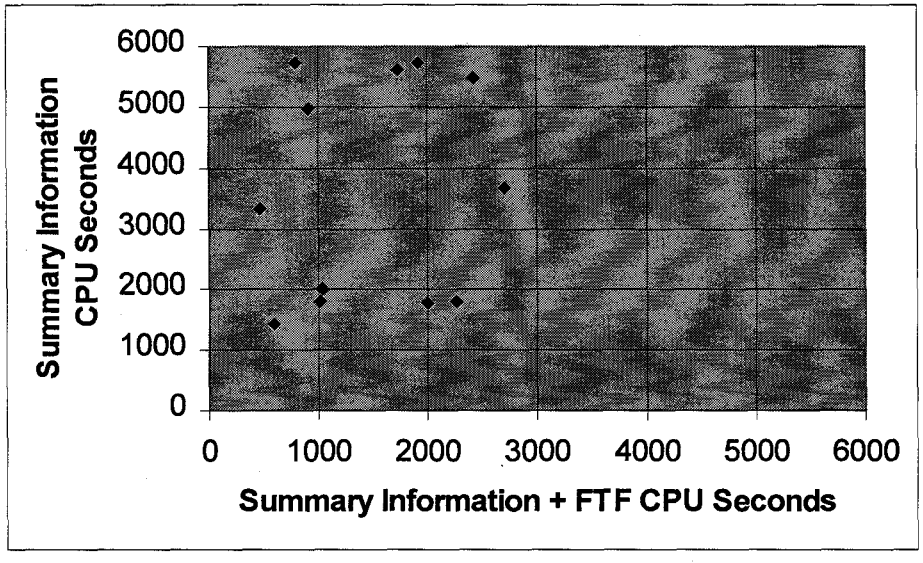
Often shared (channel) resource only

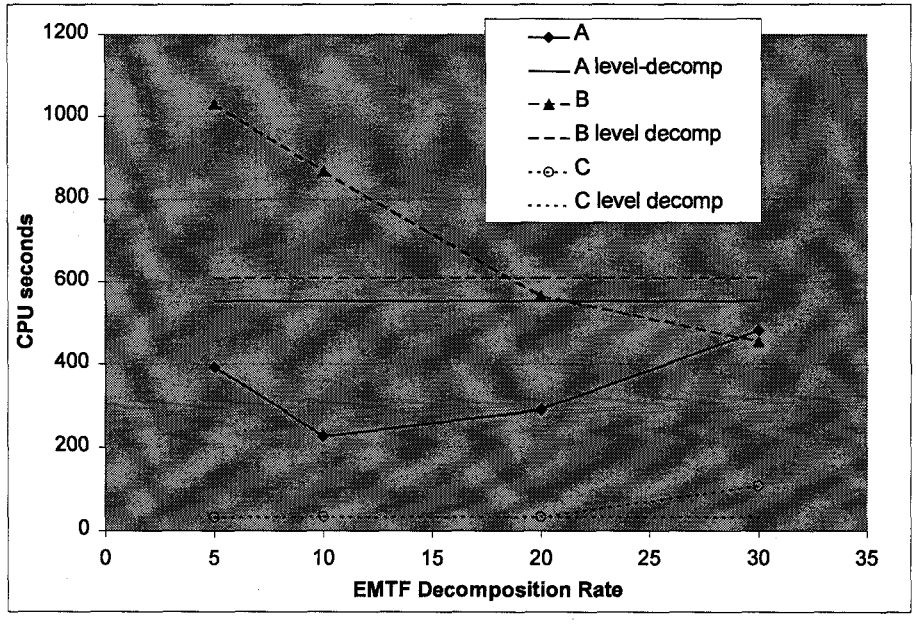
Experiments in ASPEN for a Multi-Rover Domain

- Picking branches that result in fewer conflicts (FTF) greatly improves performance.



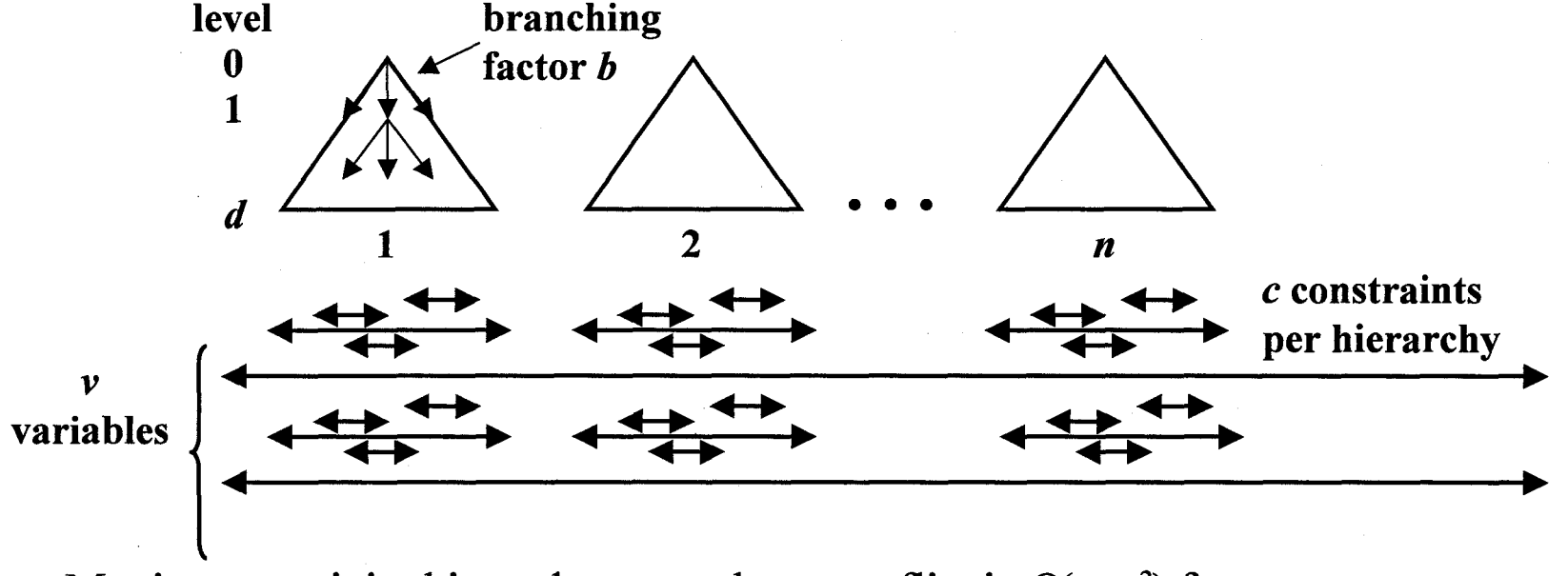
- Expanding activities involved in greater numbers of conflicts is better than level-by-level expansion when choosing a proper rate of decomposition





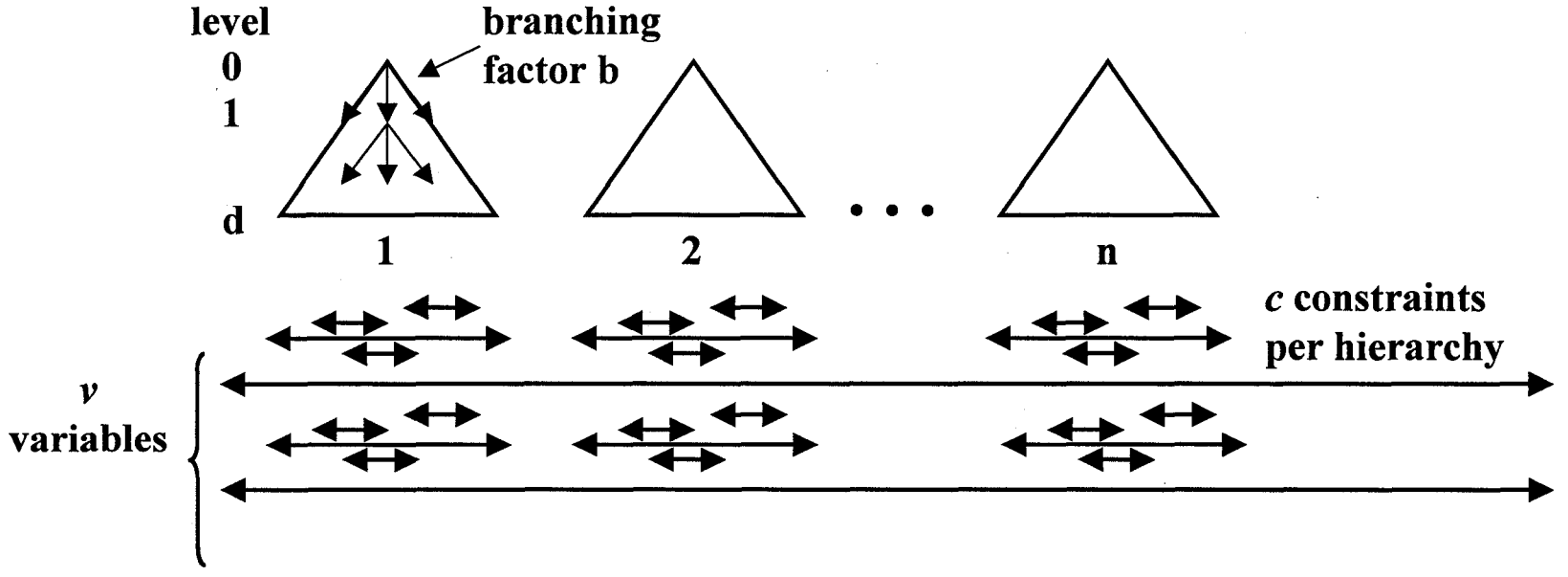
Complexity Analysis

- Iterative repair planners (such as ASPEN) heuristically pick conflicts and resolve them by moving activities and choosing alternative decompositions of abstract activities.



- Moving an activity hierarchy to resolve a conflict is $O(vnc^2)$ for v state or resource variables, n hierarchies in the schedule, and c constraints in hierarchy per variable.
- Summarization can collapse the constraints per variable making c smaller.
- In the worst case, where no constraints are collapsed because they are over different variables, the complexity of moving and activity hierarchies at different levels of expansion is the same.

Complexity Analysis



- In the other extreme, where constraints are always collapsed when made for the same variable, the number of constraints c is the same as the number of activities and grows b^i for b children per activity and depth level i. Thus, the complexity of scheduling operations grows $O(vnb^{2i})$.
- Along another dimension, the number of temporal constraints that can cause conflicts during scheduling grows exponentially ($O(b^i)$) with the number of activities as hierarchies are expanded.
- In addition, by using summary information to prune decomposition choices with greater numbers of conflicts, exponential computation is avoided.
- **Thus, reasoning at abstract levels can resolve conflicts exponentially faster.**